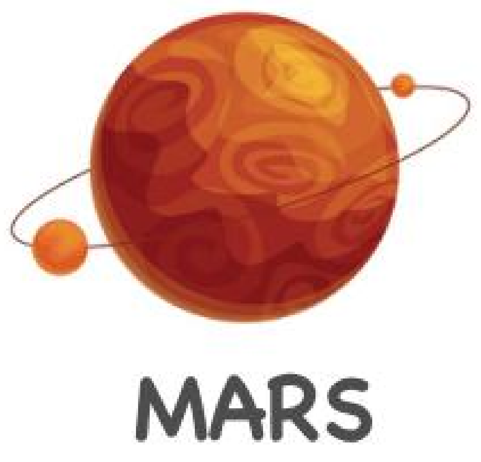
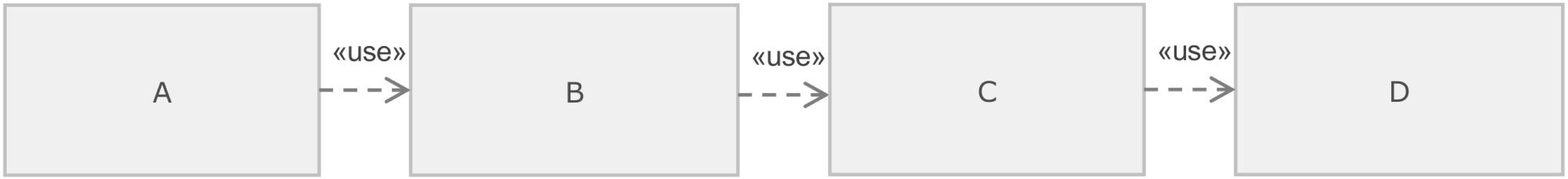
Home Exercise: A Very Small Application Core

Propose an architecture of the application core of the Mars software. It should consist of four (very small) cohesive components that use each other as shown in the diagram.



The application core is the part of the software that gets **eight** integers (e.g. D[13:91, 23:05] P[22:05, 24:45]) and returns **one** integer (the time both moons are jointly visible). Input validation is **not** part of this application core.

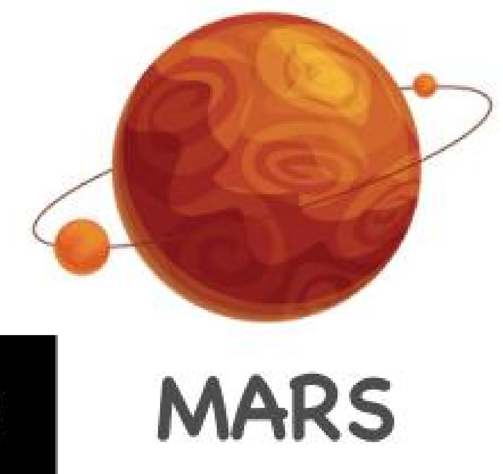


For more details, see the next two slides.

ExID: Mars02

Home Exercise: A Very Small Application Core / 2

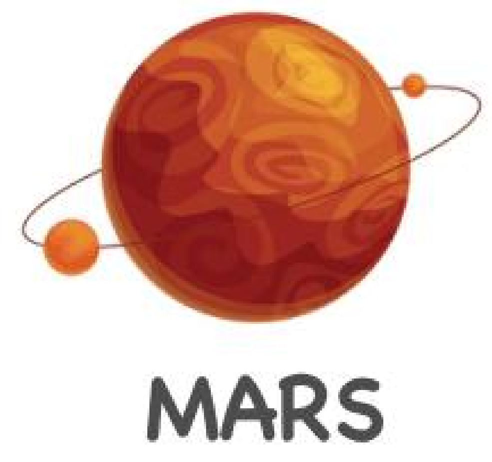
Fill out the following table. The component names should be precise and informative.



Component ID	Component name	One-sentence description of the component‘s job	Input data structure including example	Output data structure including example
A		This component ...	eight integers (e.g. D[13:91, 23:05] P[22:05, 24:45])	
B		This component …		
C		This component …		
D		This component …		one integer, namely the time both moons are jointly visible, measured in minutes, e.g. 100

Home Exercise: A Very Small Application Core / 3

Write at least two test cases for each component:



Component ID	Input data to the component	Expected output data of the component
A	D[13:91, 23:05] P[22:05, 24:45]	
A		
B		
B		
C		
C		
D		
D		

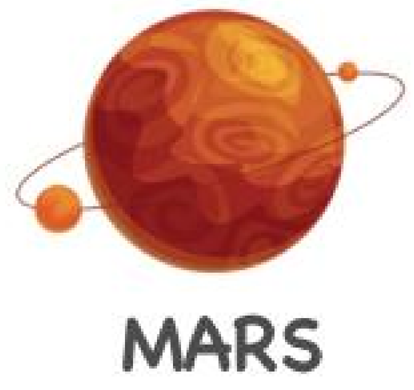
Mars: Assumptions / 2

AS2: Every day, each Mars moon rises and sets at certain points in time. Both points in time are expressed as *Mars-timestamps*, together forming a *Mars-interval*. Deimos, for instance, could rise at 4:97 and set at 12:02. That day, Phobos could rise at 24:44 and set at 7:50 the next day.

AS3: This example of [24:44, 7:50] shows that the first Mars-timestamp in a Mars-interval need not be smaller than the second Mars-timestamp. If the second one is smaller, it is about a point in time the next Mars-day.

AS4: It may be assumed that a moon rises and sets exactly in the middle of a Mars-minute.

Mars: Assumptions / 3



AS5: It may also be assumed that if a certain interval is valid for a Mars-day, it was also valid for the previous Mars-day and will be valid for the following one (although this is slightly wrong, but we can neglect the error it introduces).

AS6: On Mars, *Moon* will not be called via GUI, but by the experiment coordination software which was programmed by NASA developers.

Mars: Requirements

REQ1: Input to *Moon* should be two Mars-intervals – one for Deimos, one for Phobos.

REQ2: Output should be an integer, namely the number of Mars-minutes the intervals overlap.

For instance, D[13:91, 23:05] P[22:05, 24:45]
(D = Deimos, P = Phobos) leads to the result 100,
because there are 100 Mars minutes from 22:05 to 23:05.
Another example: D[24:53, 7:12], P[5:12, 8:45] leads to the result 200, because the intervals overlap for the 200 minutes from 5:12 to 7:12.

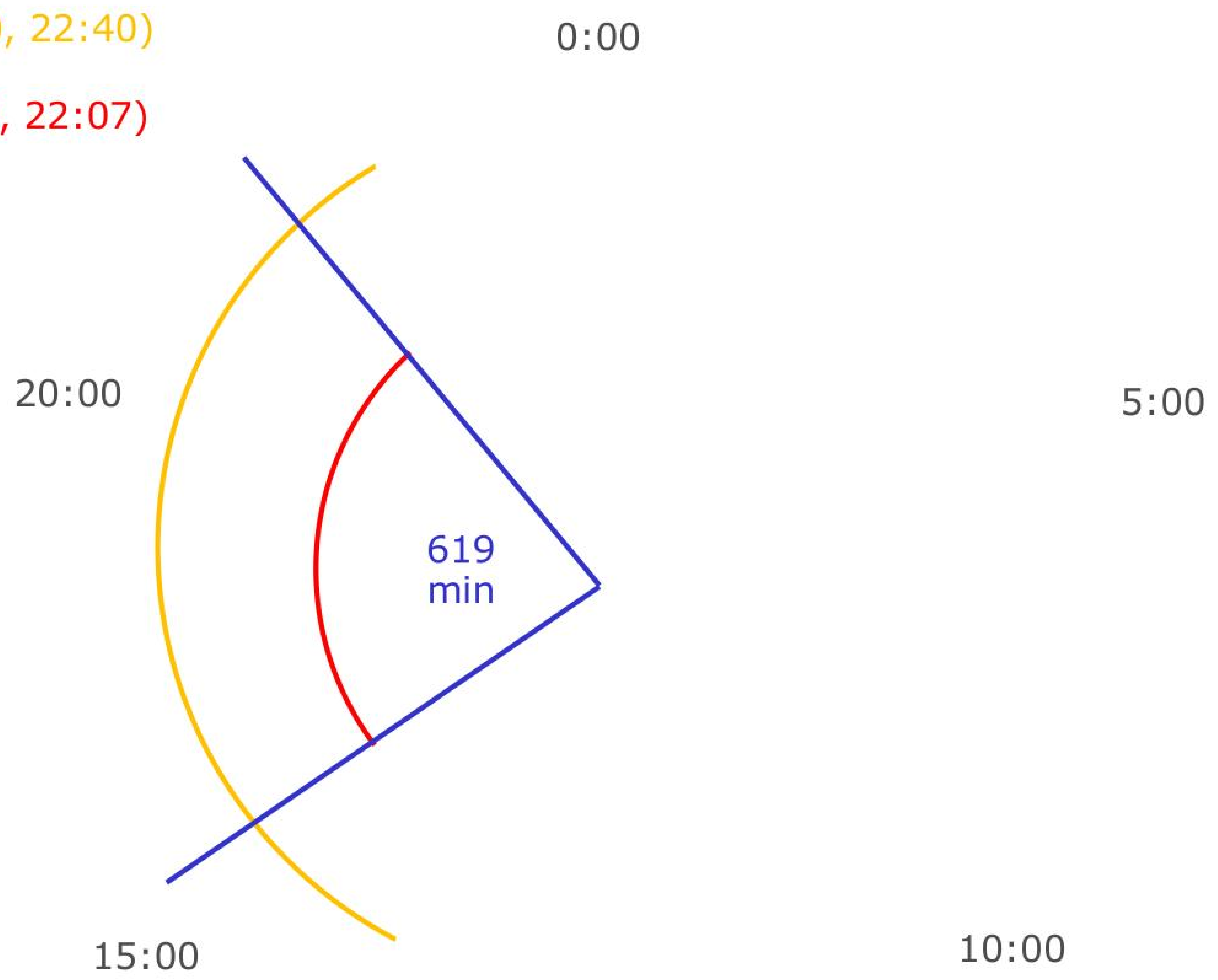
REQ3: If the two intervals have only one point in common
(example D[12:32, 17:06] P[17:06, 19:78]),
another example D[22:11, 0:36] P[7:00, 22:11])
then the result should not be zero minutes, but one minute *(„twilight rule*").

REQ4: In order to be able to test *Moon* properly **on Earth**, a human-computer interface (not necessarily a GUI) should be added. It will not be needed on Mars.

Mars: An Example

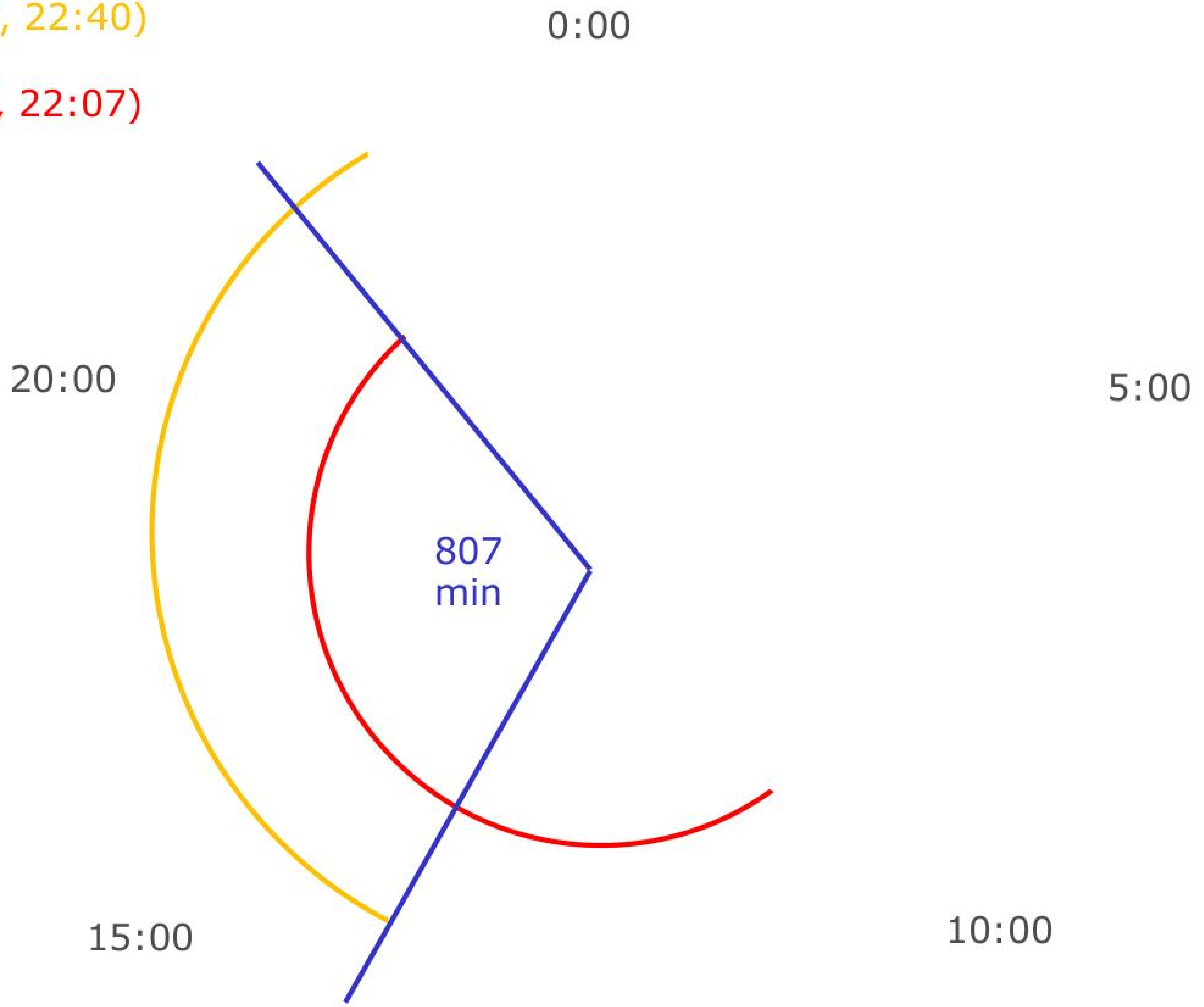
Deimos (14:00, 22:40)

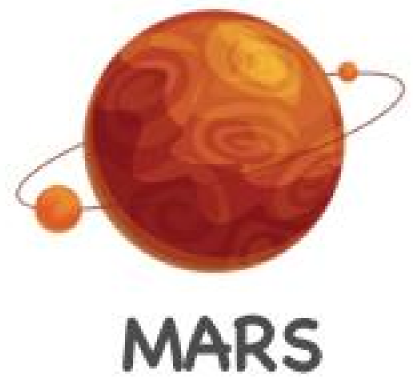
Phobos (15:88, 22:07)



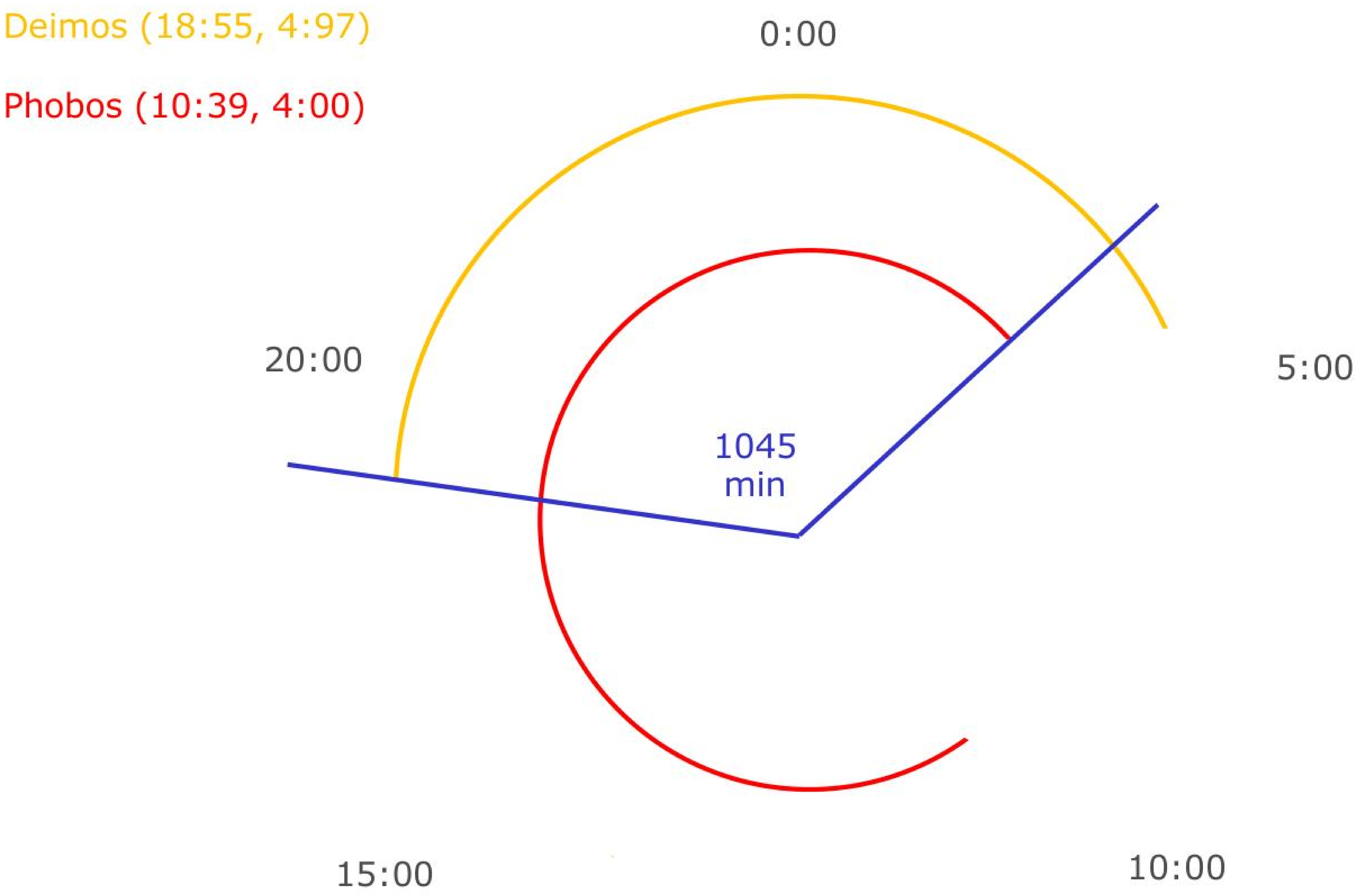
Deimos (14:00, 22:40)

Phobos (10:20, 22:07)





Mars: Yet Another Example



Deimos (18:55, 3:97)

Phobos (10:39, 18:55)